****

Visual Studio DevOps Deployment Assessment

**Deliverable Template**

DevOps Deployment Planning Services

[1 Getting the Most from Your DevOps Plan 2](#_Toc399415780)

[1.1 Give Microsoft Feedback 2](#_Toc399415781)

[2 Executive Summary 3](#_Toc399415782)

[2.1 Background 4](#_Toc399415783)

[2.2 Current Situation 6](#_Toc399415784)

[2.3 Constraints 6](#_Toc399415785)

[2.4 Recommendations 6](#_Toc399415786)

[2.5 DevOps Practices Summary 7](#_Toc399415787)

[2.6 Existing Best Practices 7](#_Toc399415788)

[2.7 Existing Hindering Practices 7](#_Toc399415789)

[2.8 Key Areas for Improvement 8](#_Toc399415790)

[2.8.1 Current State – Urgent Issues 8](#_Toc399415791)

[2.8.2 Current State – Additional Issues 8](#_Toc399415792)

[3 Roadmap to DevOps Maturity 9](#_Toc399415793)

[4 DevOps progression Framework; Using the DevOps Model 10](#_Toc399415794)

[5 Current vs. Ideal State 11](#_Toc399415795)

[6 Roadmap 12](#_Toc399415796)

[First Iteration 12](#_Toc399415797)

[Second iteration 12](#_Toc399415798)

[7 Detailed findings on DevOps Practices 14](#_Toc399415799)

[8 Resources 18](#_Toc399415800)

[9 Conclusion 21](#_Toc399415801)

1. 1. Getting the Most from Your DevOps Plan

**Guidance:**  This template includes guidance blocks and wording examples. Prior to handing over the document, remove the guidance blocks (like this one) and replace any highlighted sample text in <brackets> with your findings and recommendations.

Our recommendations for optimizing the DevOps practices and tools in your environment are detailed within this document. Please take your time to review the findings and ask any follow-up questions necessary. Depending on the capabilities of your teams and organization, you may elect to try the DevOps improvements in-house or contract with an outside consultant. In either case, this plan should be given to the party responsible for the work and used as an implementation guide.

* 1. Give Microsoft Feedback

This Planning Service has been provided as part of your Microsoft Software Assurance benefits. Please use the link below to tell Microsoft about your experience with the engagement and any improvements you would like to see made to it. The results of the survey will only be viewed by the Planning Services team at Microsoft.

[http://www.surveymonkey.com/s/dtdps\_cs](http://blogs.msdn.com/b/briankel/archive/2013/08/02/visual-studio-2013-application-lifecycle-management-virtual-machine-and-hands-on-labs-demo-scripts.aspx)

1. 2. Executive Summary

**Guidance:** The audience for this section will be interested in being able to read and digest this quickly. Keep the text in this section concise.

At the request of <Customer Name>, <Partner name> conducted a DevOps Planning Services engagement with the following objectives: <Adjust these if needed, but generally the engagement should have done most or all of these steps>

* Understand the current, complete end-to-end software lifecycle from ideation to post-production
* Elicit the desired objectives for a DevOps implementation
* Document existing Application Lifecycle Management (ALM) topology
* Create a baseline measurement of the current DevOps tools and capability
* Uncover opportunities for improvement
* Identify the most impactful areas to the business
* Document ideal end-state for teams
* Generate and present a roadmap to <implement> <improve> DevOps process using Microsoft DevOps solutions: Team Foundation Server, Visual Studio Team Services, Microsoft Azure, System Center, and others, including third-party, as appropriate

The DevOps progression Framework model was used as a framework to develop a vision and sustainable approach by which <Customer Name> can prioritize business investments that fuel business growth. The engagement focused on understanding existing development and best DevOps processes and recommending improvements. Technology, practices, and people/knowledge requirements were then identified to support the process.

* 1. Background

The following issues with the current delivery capability were articulated at the start of the assessment.

**Guidance:**

* Limited human and/or technical resources for testing
* Manual, ad-hoc, or inconsistent deployments
* Resolving production issues is slow and/or highly disruptive
* Deployments are large and take a substantial periods of time to deliver.
* High infrastructure costs and/or budget constraints
* Challenges in testing
* Lack of collaboration between IT and Development organizations

The following business priorities were articulated at the start of the assessment.

**Guidance:**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

* Team Automony and Enterprise alignment
* Rigorous management of technical debt
* Focus on flow of customer value
* Hypothesis drive development
* Evidence gathered in production
* Live site culture
* Manage infrastructure as a flexible resource

The following tools and processes (strategy) were articulated at the start of the assessment

**Guidance:** [**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://www.microsoft.com/en-us/download/details.aspx)

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

* Usage of current DevOps tools
* Challenges that exist in the DevOps tools
* Knowledge of the available Microsoft solutions and practices
  + Configuration management
  + Infrastructure as code
  + Release management
  + Continuous integration
  + Continuous deployment
  + Application performance monitoring
  + Test automation
  + Azure DevTest
  + TFS and VSTS
  + Source Control Management
  1. Current Situation

<Customer Name> is <current state of the business – growth, expansion, competition?>. The main challenges they are facing are <describe>. <High level solution> will help <Customer Name> address the following issues that were captured during the interview process:

**Guidance:** Briefly summarize the organization’s current situation and associated top issues. Identify a high-level solution.

* 1. Constraints

During the interview process at <Customer Name> a few constraints arose that would impact a DevOps adoption. Some of these are due to <Customer Name> policies and external requirements, while others are due to team or organizational issues that can be addressed in the following ways:

**Guidance:** Call out what constraint(s) might impact an adoption of DevOps practices

* <Customer Name> needs to have a high security model and workflow as the ready code moves through the environment. This requirement however does slow the delivery and testing practices as teams have a challenge accessing the environment.
* <Customer Name> has to wait for external vendors for feedback. This slows the team’s abilities to complete the story.
* <Customer Name> supports legacy applications which do not have automated tests. New unit tests cannot be added to these apps without significant refactoring.
  1. Recommendations

There are three main areas of focus for the team to improve their software development practice. Please see the Roadmap section below for a more detailed outline.

* 1. DevOps Practices Summary

**Guidance:** Call out the main practices being used. Continuous integration, Scrum or Kanban process, automated testing, Lab Management, etc.

**s**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

<Write a narrative summary of existing practices, referring to the following DevOps progression Framework categories:

Backlog

Schedule and Team

Technical Debt

Flow

Evidence

Production

Cloud

We have found that <Customer Name> is mainly performing DevOps at a <Level> level and should focus on incrementally improving.

* 1. Existing Best Practices

Our interviews surfaced the following Best Practices that are being used by teams at <Customer Name>. These practices are:

We recommend that these practices continue to be employed and are continuously evaluated and improved in order to promote process optimization.

* 1. Existing Hindering Practices

**Guidance:** The purpose of the existing hindering practices are to point out the pain points the team is experiencing from the lack of DevOps practices. The following are some examples.

**s**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://blogs.msdn.com/b/visualstudioalm/)

Our interviews surfaced a few top issues that are interfering with growth in DevOps maturity. Adopting a new mindset around these issues and prioritizing improvements in these areas will yield significant results. These issues are:

* 1. Key Areas for Improvement
     1. Current State – Urgent Issues

During our onsite interviews, we uncovered the following practices that should be considered critical and essential to improving the development capability in relation to their impact on the business. These practices were:

**Guidance: This will be the main points for the call to action**

**s**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://go.microsoft.com/fwlink/)

* + 1. Current State – Additional Issues

During the course of the assessment, we were able to identify additional issues that we believe are having a material impact on DevOps capacity and software delivery within <Customer Name>. These include:

1. Roadmap to DevOps Maturity

**Guidance:** This section will set the stage and workflow on how a company can move to an advanced maturity model. The purpose of this section is to explain that the growth and road is not a direct one and that we recommend the organizations have to have additional support as they move into different states or levels.

**s**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://www.surveymonkey.com/s/dtdps_cs)

The level of DevOps maturity varies greatly between companies and the type of software being delivered. The goals outlined in the maturity level are not “one size fits all” solutions for companies and teams; rather, they provide a direction for maturing your complete software lifecycle to meet the needs of the business and your customers.

The growth of a mature DevOps practice is not a direct one. As the team becomes more efficient, requirements become refined and testing results become more transparent, and there will be new areas of concern that will emerge.

As DevOps maturity levels move through the next stages (for example, Repeatable to Reliable), some common “growing pains” will begin to emerge. These common “growing pains” will be the best choices for additional assessments and guidance to help assist with the transition for the business, teams, and individuals.

1. DevOps progression Framework/Readiness Model

The model below shows the different practice areas and maturity levels of DevOps at a broad scale. Levels of DevOps maturity may vary across teams, even within a common department or division, and the same organization may have significantly more advanced practices in some categories than others. Enhancing the entire workflow to achieve the best possible DevOps process throughout the enterprise should be the end goal of the team and individual effort.

**Backlog**

**Schedule & Team**

**Technical Debt**

**Flow**

**Evidence**

**Production**

**Cloud**

**Foundational**

Releases: ad-hoc

Decisions: brute force

Teams: siloed

Goals: unknown

Limited or no public cloud

**Repeatable**

Releases: periodic

Decisions: anecdotal

Teams: communicative

Goals: inconsistent

Individual teams working with VM’s

**Reliable**

Releases: frequent

Decisions: corroborated

Teams: collaborative

Goals: aligned

Corporate wide with no team autonomy

**Aspirational**

Releases: continuous

Decisions: evidence-based

Teams: unified

Goals: shared

Public Cloud fully embraced

1. Current vs. Desired State

**Guidance:** This section will give an overview of the current state of the organization and how it matches with the maturity model and the organization’s Ideal State.

**s**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

The current DevOps maturity levels at <Customer Name>, on a scale of 1 (Foundational) to 4 (Aspirational), for each practice area are illustrated below. The team has some good practices in place and the enthusiasm to achieve a higher level in several key categories.

**Current Practices [<Level>]:**

**Guidance:** Give key examples here for some or all of the DevOps Progression Framework practice areas, focusing on the most impactful opportunities and/or customer’s priority goals.

**s**

[**Application Lifecycle Management with Visual Studio and Team Foundation Server**](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

**Ideal State [<Level>]:**

1. Roadmap

Based on our observations and discussions, we recommend that the following iterative roadmap be implemented in order to understand and instill DevOps best practices within the teams.

Please note that the areas for improvement mentioned in the prior section, which are marked as urgent, may not be addressed immediately. In some cases, the foundations for improving a particular service area will not be in place in the first or second iteration.

### First Iteration

The main goal of the first iteration is to identify and review the actionable items for the team and consult to review and agree on a course that would be beneficial.

#### 1. <Issue>

<Summary>

**Current issue(s) <based on above>**

**Recommendation(s)**

<Summary>

### Second iteration

Once the critical issues are in progress, we recommend having a continuous improvement progress check-in. After the first iteration, there is potential for the team to be overwhelmed with the number of new processes, tools, and changes. We thus recommend time for the team to “work on your own” after the first iteration.

In the second and subsequent iterations there will likely be additional items added to the backlog, as the team find additional pain points as they continue to work through the new processes.

#### 1. <Issue>

<Summary>

**Current issue(s) <based on above>**

**Recommendation(s)**

<Summary>

### Subsequent iterations

Repeat the cycle of improvement, retrospective, and adjustment.

1. Detailed findings on DevOps Practices

The detailed findings will indicate the maturity levels of each of the following discipline areas. The findings provide a general estimate of the maturity level and the impact it is causing. <Note: not all lines need to be filled out for each category. Focus on greater detail for a few highest-value findings and remove rows that are not useful.>

|  |  |  |
| --- | --- | --- |
| Backlog Summary : | | |
| Prioritization | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Engagement with stakeholders | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Schedule and Team Summary : | | |
| Cross-functionality | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Work in process (WIP) | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Ability to respond to schedule changes | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Technical Debt Summary : | | |
| Rework vs. new value delivery | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Flow Summary : | | |
| Build management | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Deployment process | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Evidence Summary : | | |
| Code quality | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Test automation | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Telemetry, instrumentation, ability to analyze software effectiveness in production | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Production Summary : | | |
| Detection of issues | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Responsiveness | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Scalability | | |
|  | **Maturity Observations:** |  |
|  | **Maturity Level Rating:** |  |
|  | **Impact Level Rating:** |  |
|  | **Impact Observations:** |  |
|  | **Impact Benefits:** |  |
|  | **Best Practices:** |  |
| Cloud Summary: | | |
|  | **Maturity Observations:** |
|  | **Maturity Level Rating:** |
|  | **Impact Level Rating:** |
|  | **Impact Observations:** |
|  | **Impact Benefits:** |
|  | **Best Practices:** |

1. Resources

The following is a list of very useful resources readily available to help deepen your knowledge and understanding.

#### General DevOps Training

* Microsoft Virtual Academy DevOps training

<https://mva.microsoft.com/training-topics/devops#!lang=1033>

The Microsoft Virtual Academy provides on-demand courses on DevOps

* DevOps in Azure

<https://www.microsoft.com/en-us/Openness/DevOps>

Microsofts view on openness in a DevOps organization

* IT Pro blog on DevOps in Azure

<https://azure.microsoft.com/en-us/blog/topics/it-pro-devops/>

A collection of blogs focusing on the IT Professional and all things DevOps

* DevOps in the Enterprise Blog

<https://blogs.technet.microsoft.com/devops/>

* Microsoft Open Technologies DevOps Information

https://msopentech.com/blog/project-categories/devops/

#### Testing with Visual Studio and TFS

* ALM Rangers Guide – Test Release Management:   
  <http://vsartestreleaseguide.codeplex.com/>

This Visual Studio ALM Ranger project has the primary goal of delivering scenario based and hands-on guidance for managing Microsoft Test Manager Test plans.

* ALM Rangers Guide – Visual Studio Test Tooling Guidance: [http://vsartesttoolingguide.codeplex.com/](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

This umbrella project delivers a range of practical and scenario based guidance for Visual Studio test features, such as Coded UI, Microsoft Test Manager, IntelliTrace, and Microsoft Fakes.

#### Automated Build-Deploy-Test

* Setting Up Automated Build-Deploy-Test Workflows:  
  [http://msdn.microsoft.com/en-us/library/hh191495.aspx](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

You can use a build-deploy-test workflow to deploy and test your application when you run a build. This lets you schedule and run the build, deployment, and testing of your application with one build process. Build-deploy-test workflows work with Lab Management to deploy your applications to a lab environment and run tests on them as part of the build process.

* Visual Studio Team Foundation Build Customization Guidance:  
  <http://vsarbuildguide.codeplex.com/>

This Visual Studio ALM Ranger project has the primary goal of delivering scenario based and hands-on lab guidance for the customization and deployment of Team Foundation Build.

#### Mobile Development DevOps Resources

* Xamarin Platform Overview

<http://cdn1.xamarin.com/resources/Build%20better%20apps%20with%20Xamarin%20and%20C%23.pdf>

* Xamarin Test Cloud Datasheet

<http://xamarin-releases.s3.amazonaws.com/webimages/assets/Xamarin-Test-Cloud-Datasheet.pdf>

* Xamarin Insights Datasheet

<http://cdn1.xamarin.com/resources/Improve%20your%20apps%20with%20Xamarin%20Insights.pdf>

* Whats new for Build Automation in Team Foundation Server and Visual Studio Team Services

<http://channel9.msdn.com/Events/Build/2015/3-671>

* Key Approaches for Mobile Success:   
  <http://cdn1.xamarin.com/consulting-partners/xamarin-white-paper-key-approaches-to-mobile-success.pdf>
* How to be successful with Mobile Ops

<https://blog.xamarin.com/how-to-be-successful-with-mobile-devops/>

* The Platform for Mobile Enterprise

<http://cdn1.xamarin.com/consulting-partners/Official%20Xamarin%20Deck%20-%20June2015%20(ConsultingPartners).pptx>

* Go Mobile with Xamarin and Azure

<https://www.youtube.com/watch?v=pwZoXkyK6Tk>

* Anatomy of a Native Mobile App

<http://xamarin.com/resources/native-mobile-app-white-paper>

* Fast Track your Mobile Success

<http://xamarin.com/resources/native-mobile-app-white-paper>

* Xamarin Test Cloud Overview

https://www.youtube.com/watch?v=QOZQR63yOTQ

* Get started with Xamarin Test Cloud

<https://fast.wistia.net/embed/iframe/bdmfr3hk4d?popover=true>

* HockeyApp References

<https://www.hockeyapp.net/>

* Cordova References

<https://cordova.apache.org>

#### General ALM Resources

* What’s new in Visual Studio 2015:   
  https://www.visualstudio.com/news/releasenotes/vs2015-update3-vs
* https://msdn.microsoft.com/en-us/library/bb386063.aspx
* Getting started with Application Lifecycle Management:   
  <http://msdn.microsoft.com/en-US/library/vstudio/dd286491(v=vs.110)>
* Technical Articles for Visual Studio Application Lifecycle Management  
  <http://msdn.microsoft.com/en-us/library/ee889983.aspx>
* Visual Studio ALM + Team Foundation Server Blog:   
  <https://blogs.msdn.microsoft.com/visualstudioalm/>
* ALM Rangers: TFS Planning and Disaster Avoidance and Recovery, and TFS on Azure IaaS Guide

http://vsarplanningguide.codeplex.com

* Brian Harry’s Blog: Everything you want to know about Visual Studio ALM

<http://blogs.msdn.com/b/bharry/>

* Read about “Microsoft’s Journey to Cloud Cadence” by Sam Guckenheimer

[download.microsoft.com/download/7/3/f/73fd407c-abcd-4599-90a6-90ffedfbc9e1/from agile to devops at microsoft developer division.pdf](file:///C:/Users/v-asstal/Documents/download.microsoft.com/download/7/3/f/73fd407c-abcd-4599-90a6-90ffedfbc9e1/from%20agile%20to%20devops%20at%20microsoft%20developer%20division.pdf)

* Microsoft Visual Studio Virtual Labs:

[http://msdn.microsoft.com/en-US/vstudio/ff640662](http://msdn.microsoft.com/en-us/library/fda2bad5(v=vs.120).aspx)

#### Technical ALM Resources

* DevOps and Application Lifecycle Management:

<https://www.visualstudio.com/docs/vsts-tfs-overview>

* Visual Studio 2013 Application Lifecycle Management Virtual Machine and Hands-on-Labs / Demo Scripts:

[http://blogs.msdn.com/b/briankel/archive/2013/08/02/visual-studio-2013-application-lifecycle-management-virtual-machine-and-hands-on-labs-demo-scripts.aspx](http://msdn.microsoft.com/en-us/library/hh191495.aspx)

* Team Foundation Server Migration and Integration Solutions:

<http://msdn.microsoft.com/en-us/vstudio/bb840033>

* Administer Team Foundation Server:

<http://msdn.microsoft.com/en-us/library/ms181758.aspx>

* Test Automation, Planning, and Management Concepts
* <http://vsartestreleaseguide.codeplex.com>
* Visual Studio and MSDN Licensing White Paper:  
  <https://www.microsoft.com/en-us/download/details.aspx?id=13350>
* TFS Update List of Feature in each versions:   
  <https://www.visualstudio.com/en-us/news/release-archive-vso.aspx>
* Visual Studio Capabilities Comparison: <http://www.microsoft.com/visualstudio/eng/products/compare>
* Visual Studio Forums:  
  <https://social.msdn.microsoft.com/Forums/vstudio/en-US/home>
* Visual Studio Developer Center:  
  <https://www.visualstudio.com/en-us/visual-studio-homepage-vs.aspx>

1. Conclusion

We recommend that the implementation of the practices outlined in this document be validated during the initial deployment and as projects and teams are brought on board the system. Teams constantly have to adapt and change their processes as the business and technical environment changes around them.

To encompass all of the recommendations in this document, a schedule for all of the relevant tasks should be created. Complete implementation and customization should be done by <Customer Name> operations staff.